Niveau(x) d'enseignement: 0 Numéro EDUQ : 0670360
Centre d'intérêt: 090

Brunet, Richard.

Diverses sous-routines statistiques. I / Richard Brunet. -- [Sainte-Foy] : INRS-Education, [1973?].
60 f. ; 28 cm.

Code: R-23.

Vedette(s)-secondaire(s): INRS-Education

Descripteur(s): Informatique * Programmation * Statistique * Répertoire

Identificateur(s) de sujet: FORTRAN (Langage de programmation)

Au cours de ses recherches, l'auteur a été appelé à utiliser diverses statistiques; il a parfois dû écrire des programmes FORTRAN, parfois en faire écrire et parfois en utiliser de déjà écrits. Il a donc en sa possession une série de sous-routines FORTRAN qui effectuent une gamme variée de travaux. Il présente dans ce document le matériel qu'il a en sa possession. Le document renferme donc une première série de sous-routines; elles sont présentées dans la partie B du document avec certaines de leurs spécifications. Ces sous-programmes sont provisoirement classés dans des grandes catégories dans la partie C du document.

Type de document: r

INRS

INSTITUT NATIONAL DE LA RECHERCHE SCIENTIFIQUE - UNIVERSITÉ DU QUÉBEC

EDUQ 8670360

DIVERSES SOUS-ROUTINES STATISTIQUES
(cahier no. 1)

Richard Brunet

Document 1173-01

Table des matières

		pages
A.	Présentation	1
B.	Sous-routines	3
	A.1.1 - ORDREVD	4
	A.1.2 - ORDREVC	6
	A.1.3 - ORDMLC	8
	A.2.1 - MATOUT	10
	B.1.1 - AMSIG	12
	B.1.2 - MOMENT	14
	B.1.3 - MEDIANE	16
	B.1.4 - STANDA	18
	B.2.1 - CORREL	20
	B.2.2 - FISHER	22
	B.2.3 - FISHBAK	24
	C.1.1 - INIRANF	26
	C.1.2 - ALEAT	28
	C.2.1 - SIMUL1	30
	C.2.2 - SIMUL2	32
	D.1.1 - INVSYM	34
	D.2.1 - HOW	37
	E.1.1 - VARIMAX	45
	E.1.2 - CALCV	48
	E.2.1 - BETA	50
	E.3.1 - FIDFAC	52
	E.4.1 - PREROTE	54
	E.4.2 - PLOTTRA	56
C.	Classification des sous-routines	58

A - Présentation

Au cours de mes pélerinages au sanctuaire de l'Informatique, j'ai été appelé (et le serai sans doute dans l'avenir) à utiliser diverses statistiques; j'ai parfois dû écrire des programmes FORTRAN, j'en ai parfois fait écrire et j'en ai parfois trouvé qui étaient déjà écrits.

En bref, j'ai en ma possession une série de sous-routines Fortran qui effectuent une gamme variée de travaux, ce qui me permet de me dépanner moi-même relativement rapidement.

J'ai donc pensé vous livrer dans ce document (et dans d'autres qui suivront) le matériel que j'ai en ma possession, pensant qu'il pourrait éventuellement vous être utile.

Ce document renferme une première série de sous-routines; elles sont présentées à la partie B de ce document avec certaines de leur spécifications.

Ces sous-programmes sont provisoirement classés dans des grandes catégories présentées à la partie C de ce document.

J'espère pouvoir vous fournir rapidement d'autre matériel. Vous remarquerez que les droits d'auteurs sur certaines sous-routines ne sont pas mentionnés pour la simple raison que j'aurais beaucoup de mal à dire d'où elles proviennent!

Conclusion 1: j'espère que ça vous est utile.

Conclusion 2: ce document devrait être le premier d'une série (je l'espère).

B - Les sous-routines

O R D R E V D

1. Appel

CALL ORDREVD (N, NN, A)

A: à l'entrée vecteur des données
à la sortie, ordonné par ordre décroissant
N: nombre d'éléments de N
NN: dimension de A dans le Main

2. Travail effectué

Ordonne les éléments d'un vecteur A dans un ordre décroissant (le plus grand le premier).

3. Temps d'exécution

Négligeable

4. Mémoire requise

$66_8 = 54_{10}$ mots

5. Sous-routines utilisées

Aucune

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE ORDREVD(N,NN,A)
C.....  ORDONNE LES ELEMENTS D UN VECTEUR PAR ORDRE DECROISSANT
      DIMENSION A(NN)
      NM=N-1
      DO1I=1,NM
      L=I+1
      DO1J=L,N
      IF(A(I).GT.A(J))GOTO1
      RESTE=A(I)
      A(I)=A(J)
      A(J)=RESTE
    1 CONTINUE
      RETURN$END
```

O R D R E V C

1. Appel

CALL ORDREVC (X, N, NN)

X : vecteur à trier
N : longueur de X
NN: dimension de X dans le main

2. Travail effectué

Ordonne un vecteur X par ordre croissant (le plus petit en premier).

3. Temps d'exécution

Négligeable

4. Mémoire requise

$67_8 = 55_{10}$ mots

5. Sous-routines utilisées

Aucune

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE ORDREVC(X,N,NN)
C..... ORDONNE LES ELEMENTS D UN VECTEUR PAR ORDRE CROISSANT
      DIMENSION X(NN)
      KN=N-1
      DO1I=1,KN
      L=I+1
      DO1J=L,N
      IF(X(I).LE.X(J))GOTO1
      R=X(I)
      X(I)=X(J)
      X(J)=R
    1 CONTINUESRETURNSEND
```

O R D M L C

1. Appel

CALL ORDMLC (A, N, K, N1, K1)

A : matrice à ordonner
N : nombre de lignes
K : nombre de colonnes
N1 et K1: dimension de A dans le MAIN

2. Travail effectué

Ordonne les éléments de chaque ligne de la matrice A de façon croissante (le plus petit en premier).

3. Temps d'exécution

Négligeable

4. Mémoire

$123_8 = 83_{10}$ mots

5. Sous-routines utilisées

Aucune

6. Auteur

Richard Brunet, INRS-Education

```
SUBROUTINE ORDMLC    TRACE                    CDC 6400 FTN V3.0-P336 OPT=0  73/
      SUBROUTINE ORDMLC(A,N,K,N1,K1)
C....ORDONNE LES LIGNES D UNE MATRICE PAR ORDRE CROISSANT
      DIMENSION A(N1,K1)
      DO4L=1,N
      NX=K-1
      DO5I=1,NX
      LK=I+1
      DO5J=LK,K
      IF(A(L,I).LE.A(L,J))GOTO5
      RESTE=A(L,I)
      A(L,I)=A(L,J)
      A(L,J)=RESTE
    5 CONTINUE
    4 CONTINUE$RETURN$END
```

M A T O U T

1. Mode d'appel

CALL MATOUT (R, NV, NF, KK)

R : matrice à imprimer
NV: nombre de lignes de R à imprimer
NF: nombre de colonnes de R à imprimer (maximum de 15)
KK: dimension de R dans le main

2. Travail effectué

Imprime une matrice R de NV lignes et de NF colonnes (au maximum 15).

3. Temps d'exécution

Inconnu

4. Mémoire requise

$59_{10} = 73_{B}$ mots

5. Sous-routines nécessaires

Aucune

6. Auteur

Richard Brunet, INRS-Education

CDC 6400 FTN V3.0-P336 OPT=0 75/1

SUBROUTINE MATOUT TRACE

```
      SUBROUTINEMATOUT(R,NV,NF,KK)$DIMENSIONR(KK,KK)
      PRINT1,(I,I=1,NF)
    1 FORMAT(10X15I 8/1X130(1H=))
      DO2I=1,NV
    2 PRINT3,I,(R(I,J),J=1,NF)
    3 FORMAT(I10,15F 8.2)
      RETURN$END
```

CDC 6400 FTN V3.0-P336 OPT=0

61 JCDC 6400 FTN V3.0-P336 OPT=0

MAPTRACRELOCATION

AMSIG

1. Appel

CALL AMSIG (N, NN, X, S, A)

N : longueur du vecteur
NN: dimension de A dans le MAIN
X : moyenne
S : sigma
A : vecteur

2. Travail effectué

Calcule la moyenne et le sigma de la distribution de A et retourne ces valeurs dans X et S respectivement.

3. Temps d'exécution

Négligeable

4. Mémoire

$103_8 = 67_{10}$ mots

5. Sous-routines nécessaires

FLOAT intrinsèque
SQRT intrinsèque

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE AMSIG(N,NN,X,S,A)
C       CALCULE LA MOYENNE ET LE SIGMA D UN VECTEUR
      DIMENSIONA(NN)
      AN=FLOAT(N)
      X=S=0.
      DO200I=1,N
      X=X+A(I)
  200 S=S+A(I)*A(I)
      S=SQRT((S-X*X/AN)/(AN-1.))
      X=X/AN
      RETURNSEND
```

M O M E N T

1) Appel

CALL MOMENT (N, A, NN)

A : vecteur des données
N : longueur de A
NN: dimension de A dans le Main

2) Travail effectué

Etant donné un vecteur A, on trouve la moyenne, le sigma, la symétrie et la voussure de la distribution. Ces valeurs sont imprimées.

3) Temps d'exécution

Négligeable

4) Mémoire requise

$174_8 = 126_{10}$ mots

5) Sous-routines utilisées

FLOAT intrinsèque

6) Auteur

Richard Brunet, INRS-Education

```
BROUTINE  MOMENT    TRACE                          CDC 6400 FTN V3.0-P336 OPT=0   73/
            SUBROUTINE MOMENT(N,A,NN)
            DIMENSION A(NN)
      C   CALCULE LA MOYENNE, LE SIGMA,LA SYMMETRIE ET LA VOUSSURE DE LA DISTRI-
      C  BUTION DE A..........
            G1=G2=G3=G4=0.
            DO11I=1,N
            Z=A(I)
            G1=G1+Z
            G2=G2+Z*Z
            G3=G3+Z*Z*Z
            G4=G4+Z*Z*Z*Z
          1 CONTINUE
            AN=FLOAT(N)
            AM=G1/AN$G2=G2/AN$G3=G3/AN$G4=G4/AN
            VA=G2-AM*AM
            G1=G3-3.*AM*VA+2.*AM*AM*AM
            G2=G4-4.*AM*G3+6.*AM*AM*VA-3.*AM*AM*AM*AM
            G1=G1/(VA**1.5)
            G2=G2/(VA*VA)-3.
            PRINT2,AM,VA,G1,G2
          2 FORMAT(//10X*MOYENNE=*F10.3/12X*SIGMA=*F10.3/8X*SYMMETRIE=*F10.3/
           19X*VOUSSURE=*F10.3)
            RETURN$END
```

M E D I A N E

1. Appel

CALL MEDIANE (X, N, NDIM, Y, XMD)

X : vecteur dont il faut trouver la médiane
N : longueur de X
Ndim: dimension de X et Y dans le main
Y : vecteur de travail
XMD : valeur de retour pour la médiane

2. Travail effectué

Trouve la médiane du vecteur X et retourne la valeur dans XMD.

3. Temps d'exécution

Négligeable

4. Mémoire

5. Sous-routines utilisées

ORDREVC
MOD intrinsèque

6. Auteur

Richard Brunet, INRS-Education

SUBROUTINE MEDIANE TRACE CDC 6400 FTN V3.0-P336 OPT=0 73/

```
      SUBROUTINE MEDIANE(X,N,NDIM,Y,XMD)
      DIMENSIONX(NDIM),Y(NDIM)
      DO11=1,N
    1 Y(I)=X(I)
      CALLORDREVC(Y,N,NDIM)
      IF(MOD(N,2).EQ.0)GOTO2
      K=N/2+1
      XMD=Y(K)$RETURN
    2 K=N/2
      L=K+1
      XMD=(Y(K)+Y(L))/2.
      RETURN$END
```

S T A N D A

1. Appel

CALL STANDA (N, NN, A)

N : longueur du vecteur A
NN: dimension de A dans le MAIN
A : vecteu : à standardiser

2. Travail effectué

Standardise les éléments d'un vecteur A à une moyenne de 0. et un sigma de 1.

3. Temps d'exécution

Négligeable

4. Mémoire requise

$55_8 = 45_{10}$ mots

5. Sous-routines nécessaires

AMSIG

6. Auteur

Richard Brunet, INRS-Education

```
SUBROUTINE  STANDA    TRACE                    CDC 6400 FTN V5.0-P556 OPT=0   75/
      SUBROUTINE STANDA(N,NN,A)
C  STANDARDISE UN VECTEUR A MOY=0 ET SIGMA=1.0
      DIMENSION A(NN)
      CALL AMSIG(N,NN,X,S,A)
      DO 300 I=1,N
  300 A(I)=(A(I)-X)/S
      RETURN$END
```

C O R R E L

1. Appel

CALL CORREL (N, X, Y, R, NN)

N : longueur des vecteurs X et Y
X et Y: vecteurs d'entrée
R : corrélation entre X et Y (sortie)
NN: dimension de X et Y dans le main

2. Travail effectué

Calcule la corrélation entre les éléments de X et Y et la retourne dans R.

3. Temps d'exécution

4. Mémoire requise

$132_8 = 90_{10}$ mots

5. Sous-routines utilisées

FLOAT intrinsèque
SQRT intrinsèque

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE CORREL(N,X,Y,R,NN)
C  CALCULE LA CORRELATION ENTRE DEUX VECTEURS X ET Y DE DONNEES
      DIMENSION X(NN),Y(NN)
      SX=SY=SXY=SXX=SYY=0.
      DO1I=1,N
      SX=SX+X(I)$SY=SY+Y(I)
      SXX=SXX+X(I)*X(I)$SYY=SYY+Y(I)*Y(I)
    1 SXY=SXY+X(I)*Y(I)
      AN=FLOAT(N)
      ANUM=AN*SXY-SX*SY
      R=SQRT((AN*SXX-SX*SX)*(AN*SYY-SY*SY))
      R=ANUM/R
      RETURN$END
```

F I S H E R

1. Appel

   y = Fisher (R)

   R = corrélation

2. Travail effectué

   Transforme R en Z de Fisher; si r est négatif, le Z sera également négatif.

3. Temps d'exécution

   Négligeable

4. Mémoire requise

   $46_8 = 38_{10}$

5. Sous-routines utilisées

   ABS intrinsèque
   ALOG intrinsèque

6. Auteur

   Richard Brunet, INRS-Education

```
UNCTION    FISHER    TRACE                    CDC 6400 FTN V5.0-P556 OPT=0  75/
      FUNCTION FISHER(X)
C..... TRANSFORME UN R EN Z DE FISHER
      FISHER=0.
      IF(X.EQ.0.)RETURN
      FISHER=ABS(X)
      FISHER=ALOG((1.+FISHER)/(1.-FISHER))/2.
      IF(X.LT.0.)FISHER=-FISHER
      RETURNSEND
```

F I S H B A K

1. Appel

   R = FISHBAK (Z)

   Z = Z de Fisher

2. Travail effectué

   Transforme un Z de Fisher en corrélation; si Z est négatif, R sera négatif.

3. Temps d'exécution

   Négligeable

4. Mémoire requise

   $45_8 = 37_{10}$ mots

5. Sous-routines utilisées

   ABS intrinsèque
   EXP intrinsèque

6. Auteur

   Richard Brunet, INRS-Education

```
      FUNCTION FISHBAK(X)
C..... TROUVE LE R CORRESPONDANT A UN Z DE FISHER
      FISHBAK=0.
      IF(X.EQ.0.)RETURN
      FISHBAK=EXP(2.*ABS(X))
      FISHBAK=(FISHBAK-1.)/(FISHBAK+1.)
      IF(X.LT.0.)FISHBAK=-FISHBAK
      RETURNSEND
```

I N I R A N F

1. Appel

CALL INIRANF (DUM)

DUM: argument symbolique nécessité par la syntaxe, mettre la valeur 0.0

2. Travail effectué

Initie la fonction RANF (nombre aléatoire) de sorte que la même série de nombre n'apparait pratiquement pas ? fois!

3. Temps d'exécution

Inconnu

4. Mémoire requise

$102_8 = 66_{10}$ mots

5. Sous-routines nécessaires

SECOND: intrinsèque
TIME : intrinsèque
DATE : intrinsèque
RANSET: intrinsèque

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE INIRANF(DUM)
C  INITIALISE RANF AU HASARD
      CALL SECOND(X)$CALL TIME(NTIME)$CALL DATE(NDATE)
      DECODE(10,1,NTIME)T1,T2,T3
      DECODE(10,3,NDATE)D1,D2,D3
    3 FORMAT(1X3(F2.0,1X))
    1 FORMAT(3(F2.0,1X))
      T1=(T1*T2)-T3
      D1=(D1*D2)/D3
      T1=(T1/D1)+X
      CALL RANSET(T1)
      RETURN$END
```

A L E A T

1. Appel

CALL ALEAT (N, K, A, NN)

N : nombre de nombres à créer
K : chaque nombre est la moyenne de K nombres aléatoires
A : vecteur où placer les nombres
NN: dimension de A dans le main

2. Travail effectué

Crée N nombres aléatoires dont la distribution est normale. Ces nombres, emmagasinés dans A, ont une moyenne de 0 et un sigma de 1. Chaque nombre est constitué par la moyenne de K nombres aléatoires ayant une distribution rectangulaire.

3. Temps d'exécution

Pour créer 5000 nombres, chacun basé sur 100 nombres aléatoires: $\approx$ 20 secondes

4. Mémoire requise

$103_8 = 67_{10}$ mots

5. Sous-routines requises

FLOAT intrinsèque
RANF intrinsèque
INIRANF
STANDA

6. Auteur

Richard Brunet, UNRS-Education

```
SUBROUTINE ALEAT     TRACE                    CDC 6400 FTN V3.0-P336 OPT=0   73/
      SUBROUTINE ALEAT(N,K,A,NN)
C     GENERE N NOMBRES ALEATOIRES NORMAUX ENTRE 0 ET 1
      DIMENSION A(NN)
      CALL INITRANF(0)
      AK=FLOAT(K)
      DO100I=1,N
      X=0.
      DO50J=1,K
      Z=RANF(DUM)
      X=X+Z
   50 CONTINUE
      A(I)=X/AK
  100 CONTINUE
      CALL STANDA(N,NN,A)
      RETURN$END
```

S I M U L 1

1. Appel

CALL SIMUL1 (N, X, Y, NN, R)

N : longueur des vecteurs X et Y
X : vecteur d'entrée et de sortie
Y : vecteur de sortie
NN: dimension de X et Y dans le main
R : corrélation désirée entre X et Y

2. Travail effectué

Génère, à partir des éléments de X un vecteur Y tel que la corrélation attendue entre X et Y sera égale à R spécifié à l'entrée; l'erreur à une distribution normale.

3. Temps d'exécution

4. Mémoire requise

$100_8 = 64_{10}$ mots

5. Sous-routines nécessaires

RANF intrinsèque
SQRT intrinsèque
INIRANF

6. Auteur

Richard Brunet, INRS-Education

```
SUBROUTINE SIMUL1    TRACE                    CDC 6400 FTN V3.0-P336 OPT=0  75/
      SUBROUTINE SIMUL1(N,X,Y,NN,R)
C     A PARTIR D,UN VECTEUR STANDARDISE X, CREATION D UN VECTEUR Y DE CORREL
C     R DONNEE ENTRE LES DEUX
C  LE TERME D ERREUR A UNE DISTRIBUTION NORMALE
      DIMENSION X(NN),Y(NN)
      ERREUR=SQRT((1.-R*R)/(R*R))
      CALL INIRANF(0)
      DO1I=1,N
      L=RANF(0)*N+1
    1 Y(I)=X(I)+ERREUR*X(L)
      RETURNSEND
```

S I M U L 2

1. Appel

CALL SIMUL2 (N, X, Y, NN, R)

N : nombre de nombres à créer
X : vecteur d'entrée
Y : vecteur de sortie où seront les nombres
NN: dimension de X et Y dans le MAIN
R : niveau de corrélation entre X et Y

2. Travail effectué

Génère à partir de X un vecteur Y tel que la corrélation attendue entre X et Y sera égale à R spécifié à l'entrée; l'erreur aura une distribution rectangulaire.

3. Temps d'exécution

A peu près identique à SIMUL1... négligeable

4. Mémoire requise

5. Sous-routines utilisées

RANF intrinsèque
SQRT intrinsèque
INIRANF

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE SIMUL2(N,X,Y,NN,R)
C.... A PARTIR D UN VECTEUR X, CREE UN VECTEUR Y DE CORRELATION R ENTRE LE
C.... LE TERME D ERREUR A UNE DISTRIBUTION RECTANGULAIRE
      DIMENSION X(NN),Y(NN)
      ERREUR=SQRT((1.-R*R)/(R*R))
      CALL INIRANF(0)
      DO1I=1,N
    1 Y(I)=X(I)+ERREUR*((RANF(0)-0.501424)/0.288807)
      RETURNSEND
```

I N V S Y M

1. Mode d'appel

CALL INVSYM (DET, NV, A, SPVT, NN)

DET : déterminant de la matrice
NV : nombre de variables dans la matrice
A : matrice à inverser
SPVT: vecteur de travail
NN : dimension de SPVT et A dans le MAIN

2. Travail effectué

Inversion d'une matrice symétrique par la méthode de Gauss-Jordan. La matrice inversée est retournée dans A (triangle inférieur et diagonale). Le triangle supérieur de A reste inchangé.

3. Temps d'exécution

Matrice de 65 x 65: 11 secondes

4. Mémoire requise

$250_{10} = 372_8$ mots

5. Sous-routines nécessaires

Aucune

6. Auteur

Inconnu, adapté par Normand Daigle, INRS-Education

```
SUBROUTINE  INVSYM    TRACE                    CDC 6400 FTN V3.0-P336 OPT=0   75/1
      SUBROUTINE INVSYM(DET,NV,A,SPVT,NN)
      DIMENSION SPVT(NN),A(NN,NN)
      DET=1.0
      NM1=NV-1
      DO10I=1,NV
   10 SPVT(I)=1.0
      DO120I=1,NM1
      XMAX=0.0
      DO20J=1,NM1
      IF(SPVT(J))20,20,15
   15 IF(A(J,J).LE.XMAX)GOTO20
      XMAX=A(J,J)
      II=J
   20 CONTINUE
      IF(XMAX.GT.1E-10)GOTO40
      PRINT25,I
   25 FORMAT(/20(1H*)*ERREUR--MATRICE SINGULIERE,  RANG=*I3,20(1H*))
      DO30J=1,NV
      IF(SPVT(J).GT.0.)A(J,J)=1E+10
   30 CONTINUE
      DET=.5E-10
      RETURN
   40 PIV=A(II,II)
      DET=DET*PIV
      A(II,II)=1.0/PIV
      IF(II.EQ.1)GOTO70
      IJ=II-1
      DO60K=1,IJ
      CUM=A(II,K)/PIV
      DO50J=K,NV
      IF(J-II)42,45,48
   42 A(J,K)=A(J,K)-CUM*A(II,J)*SPVT(J)
      GOTO50
   45 A(II,K)=CUM
      GOTO50
   48 A(J,K)=A(J,K)-CUM*A(J,II)
   50 CONTINUE
   60 CONTINUE
      IF(II.EQ.NV)GOTO100
   70 IK=II+1
```

```
      DO90K=IK,NV
      XMAX=-A(K,II)/PIV
      CUM=XMAX*SPVI(K)
      DO80J=K,NV
   80 A(J,K)=A(J,K)+A(J,II)*CUM
   90 A(K,II)=XMAX
  100 SPVI(II)=-1.0
  120 CONTINUE
      RETURNSEND
```

H O W

1. Mode d'appel

CALL HOW (MVAR, MDIM, NVECT, R, E, V, A, B, C, D)

MVAR: ordre de la matrice de corrélation R
MDIM: dimension de R, E, V, A, B, C, D dans le main
NVEC: nombre de composantes principales à extraire
R : matrice de corrélation
E : vecteur de retour pour les valeurs-propres
V : matrice des vecteurs-propres
A, B, C, D: vecteurs de travail

2. Travail effectué

Trouve les valeurs-propres et les vecteurs propres d'une matrice symétrique selon l'algorithme de Householder, Ortega et Wilkinson.

Référence: Cooley & Lohnes

3. Temps d'exécution

Inconnu

4. Mémoire requise

$1006_{10} = 1756_{8}$ mots

5. Sous-routines requises

Aucune

6. Auteur

Cooley, W.W., Lohnes, P.R. (1971) Multivariate data analysis. New York: Wiley.

```
SUBROUTINE HOW       TRACE                    CDC 6400 FTN V3.0-P336 OPT=0   73/1
      SUBROUTINE HOW(MVAR,MDIM,NVECT,R,E,V,A,B,C,D)
      DIMENSIONR(1),E(1),V(1),A(1),B(1),C(1),D(1)
C     EIGENVALUES AND EIGENVECTORS OF A SYMMETRIC MATRIX, ALGORITHM
C        BY HOUSEHOLDER, ORTEGA, AND WILKINSON.  ORIGINAL PROGRAM BY
C     DAVID H. MATULA UNDER THE DIRECTION OF WILLIAM MEREDITH, UNIVERSITY
C        OF CALIFORNIA, BERKELEY, 1962.
C     MODIFIED BY P. R. LOHNES, PROJECT TALENT, 1966.
C
C     M IS THE ORDER OF THE INPUT MATRIX, R.
C     MD IS THE DIMENSIONED SIZE OF R IN THE MAIN PROGRAM.
C     NV IS THE NUMBER OF EIGENVECTORS TO BE COMPUTED.
C     E IS THE VECTOR IN WHICH THE EIGENVALUES ARE RETURNED.
C     V IS THE MATRIX IN WHICH THE EIGENVECTORS ARE RETURNED.
C        THE EIGENVECTORS ARE STORED AS COLUMNS IN V.
C     A,B,C, AND D ARE WORKSPACE VECTORS.
C
      EQUIVALENCE (S1,IS1),(S2,IS2)
C
      M = MVAR
      MD = MDIM
      NV = NVECT
      IF(M-1)  100, 97, 96
   96 M1 = M - 1
C     TRI-DIAGONALIZE THE MATRIX.
      M2 = M1 * MD + M
      M3 = M2 - MD
      M4 = MD + 1
      L = 0
      DO 1  I = 1, M2, M4
      L = L + 1
    1 A(L) = R(I)
      B(1) = 0.0
      IF (M - 2) 13, 2, 3
    3 KK = 0
      DO 15 K = 2, M1
      KL = KK + K
      KU = KK + M
      KJ = K + 1
      SUM = 0.0
      DO 4  J = KL, KU
```

```
SUBROUTINE  HOW       TRACE                          CDC 6400 FTN V3.0-P336 OPT=0  75/10
            4         SUM = SUM + R(J) **2
                      S = SQRT(SUM)
                      Z = R(KL)
                      B(K)=SIGN(S,-Z)
                      S = 1.0 / S
                      C(K) = SQRT (ABS (Z) * S + 1.0)
                      X = SIGN (S / C(K), Z)
                      R(KL) = C(K)
                      DO 5 I = KJ,M
                      JJ = I + KK
                      C(I) = X * R(JJ)
            5         R(JJ) = C(I)
                      DO 8  J = K, M
                      JJ = J - 1
                      D(J) = 0.0
                      L = KK + J
                      DO 6  I = K, J
                      L = L + MD
            6         D(J) = D(J) + R(L) * C(I)
                      IF (JJ - M) 7, 7, 9
            7         DO 8  I = JJ, M
                      L = L + 1
            8         D(J) = D(J) + R(L) * C(I)
            9         X = 0.0
                      DO 10  J = K, M
            10        X = X + C(J) * D(J)
                      X = .50 * X
                      DO 11  I = K, M
            11        D(I) = X * C(I) - D(I)
                      LL = KK
                      KK = KK + MD
                      DO 15  I = K, M
                      LL = LL + MD
                      DO 15  J = I, M
                      L = LL + J
            15        R(L) = R(L) + D(I) * C(J) +D(J) * C(I)
                      L = 1
                      DO 12  I = 1, M
                      X = A(I)
                      A(I) = R(L)
```

```
      R(L) = X
   12 L = L + M4
    2 B(M) = R(M3)
C     COMPUTE EIGENVALUES.
   13 BD = ABS (A(1))
      DO 14  I = 2, M
   14 BD = AMAX1(BD, ABS (A(I)) + B(I)**2)
      BD = BD + 1.0
      DO 16  I = 1, M
      A(I) = A(I) / BD
      B(I) = B(I) / BD
      D(I) = 1.0
   16 E(I) = -1.0
      DO 37  K = 1, M
   17 IF((D(K)-E(K))/AMAX1(ABS (D(K)),ABS(E(K)),1.0E-9)-1.0E-6) 37,37,18
   18 X = (D(K) + E(K)) * .50
      IS2 = 1
      C(1) = A(1) - X
      IF (C(1)) 19, 20, 20
   19 IS1 = -1
      N = 0
      GO TO 21
   20 IS1 = 1
      N = 1
   21 DO 31  I = 2, M
      IF (B(I))  22, 26, 22
   22 IF (B(I-1))  23, 27, 23
   23 IF(ABS (C(I-1))+ABS (C(I-2))-1.0E-15) 24, 25, 25
   24 C(I-1) = C(I-1) * 1.0E15
      C(I-2) = C(I-2) * 1.0E15
   25 C(I)=(A(I)-X)*C(I-1)-B(I)**2*C(I-2)
      GO TO 28
   26 C(I) = (A(I)-X) * SIGN (1.0, S1)
      GO TO 28
   27 C(I) = (A(I)-X) * C(I-1) - SIGN (B(I)**2, S2)
   28 S2 = S1
      IF (C(I))  29, 30, 29
   29 S1 = SIGN (S1, C(I))
      IF (IS2+IS1)  30, 31, 30
   30 N = N + 1
```

```
      31   CONTINUE
           N = M - N
           IF (N - K)  34, 32, 32
      32   DO 33  J = K, N
      33   D(J) = X
      34   N = N + 1
           IF (M - N)  17, 35, 35
      35   DO 36  J = N, M
           IF (X - E(J))  17, 17, 36
      36   E(J) = X
           GO TO 17
      37   CONTINUE
           DO 38  I = 1,M
            A(I) = A(I) * BD
           B(I) = B(I) * BD
      38   C(I) = (D(I) + E(I)) * BD * .50
           M1 = M
           K = 1
      39   I = 1
      40   DO 43  J = 1, M1
           IF (I - J) 41, 43, 41
      41   IF (C(I) - C(J))  43, 43, 42
      42   I = J
           GO TO 40
      43   CONTINUE
           E(K) = C(I)
           K = K + 1
           M1 =M1 - 1
           IF (I - M1 - 1)  44, 46, 46
      44   DO 45 M2 = 1, M1
      45   C(M2) = C(M2+1)
      46   IF (M1 - 1)  47, 47, 39
      47   E(K) =C(1)
           IF (ISIGN (1, NV)) 79, 76, 76
      76   DO 77  I = 1, M
      77   C(I) = E(I)
           J=M
           DO 78  I = 1,M
           E(I) =C(J)
      78   J = J - 1
```

```
 79   CONTINUE
C
C     DECIDE WHETHER TO COMPUTE EIGENVECTORS, AND IF SO, HOW MANY
C
      IF (NV)  48, 99, 48
 48   KX = IABS(NV)
      J = 1
      DO 98  INV = 1, KX
      X = A(1) - E(INV)
      Y = B(2)
      M1 = M - 1
      DO 54  I = 1, M1
      IJ = J + I-1
      IF (ABS (X) - ABS (B(I+1))) 49, 51, 53
 49   C(I) = B(I+1)
      D(I) =A(I+1) - E(INV)
      V(IJ) = B(I+2)
      Z = -X / C(I)
      X = Z * D(I) + Y
      IF (M1 - I)  50, 54, 50
 50   Y = Z * V(IJ)
      GO TO 54
 51   IF (X)  53, 52, 53
 52   X = 1.0E-10
 53   C(I) = X
      D(I) = Y
      V(IJ) = 0.0
      X = A(I+1) - (B(I+1) / X * Y + E(INV))
      Y = B(I+2)
 54   CONTINUE
      MJ = M + J - 1
      IF (X)  56, 60, 56
 56   V(MJ) = 1.0 / X
 57   I = M1
      IJ = J + I - 1
      V(IJ) = (1.0 - D(I) * V(MJ))/ C(I)
      X = V(MJ)**2 + V(IJ)**2
 58   I = I - 1
      IJ = J + I - 1
      IF (I)  59, 61, 59
```

```
SUBROUTINE  MDW       TRACE                                CDC 6400 FTN V3.0-P336 OPT=0  73/1
          59  V(IJ) = (1.0 - (D(I) * V(IJ+1) + V(IJ) * V(IJ + 2))) / C(I)
              X = X + V(IJ)**2
              GO TO 58
          60  V(MJ) = 1.0E10
              GO TO 57
          61  X = SQRT(X)
              DO 62  I = 1, M
              IJ = J +I - 1
          62  V(IJ) = V(IJ) /X
              J1 = M1 * MD - MD
              K = M
              GO TO 66
          63  K = K - 1
              J1 = J1 - MD
              Y =0.0
              DO 64  I = K, M
              IJ = J + I - 1
              L = J1 + I
          64  Y = Y + V(IJ) * R(L)
              DO 65   I = K,M
              IJ = J + I - 1
              L = J1 + I
          65  V(IJ) = V(IJ) - Y * R(L)
          66  IF (J1)  63, 67, 63
          67  NPLUS = 0
              NMIN = 0
              DO 70  I = 1, M
              IJ = J + I - 1
              IF (V(IJ))  68, 69, 69
          68  NMIN = NMIN + 1
              GO TO 70
          69  NPLUS = NPLUS + 1
          70  CONTINUE
              IF (NPLUS - NMIN)  71, 73, 73
          71  DO 72  I = 1, M
              IJ = J + I - 1
          72  V(IJ) =-V(IJ)
          73  CONTINUE
          98  J = J + MD
          C   RESTORE THE INPUT MATRIX.
```

```
      99    MD1 = MD + 1
            JJ = MD1
            M1 = M * MD
            DO 75  I = 2, M1, MD1
            K = 1
            DO 74   J = JJ, M1, MD
            R(K) = R(J)
      74    K = K + 1
      75    JJ = JJ + MD1
            GO TO 100
      97    E(1) = R(1)
            V(1) = 1.0
     100    RETURN
            END
```

V A R I M / X

1. Mode d'appel

CALL VARIMAX (NV, NF, ND, R, S, A, H, B)

NV: nombre de variables
NF: nombre de facteurs sur lesquels effectuer la rotation
ND: dimensions de R, S, A, H et B dans le MAIN
R : matrice carrée des composantes principales
S : vecteur de travail
A : vecteur de travail
H : vecteur de travail; à la sortie, contient les communautés
B : matrice de travail

2. Travail effectué

A partir de la matrice des composantes principales (R), NF facteurs sont conservés et normalisés, les communautés sont calculées, les rotations sont effectuées (processus itératif), la matrice de structure est dénormalisée et les résultats suivants sont imprimés:

1) proportion de la variance totale expliquée par chaque facteur

2) saturations maximales sur chaque facteur

3) communautés des variables

4) matrice de structure factorielle

3. Temps d'exécution

Inconnu

4. Mémoire requise

$562_{10} = 1062_8$ mots

5. Sous-routines nécessaires

1) CALCV

2) MATOUT

6. Auteur

Inconnu, Université du Texas. Adapté par Richard Brunet, INRS-Education.

```
SUBROUTINE  VARIMAX   TRACE                                  CDC 6400 FTN V3.0-P336 OPT=0  73/10

      SUBROUTINE VARIMAX(NV,NF,ND,R,S,A,H,B)
      DIMENSION R(ND,ND),S(ND),A(ND),H(ND),B(ND,15)
      READ(IND,2)((R(I,J),J=1,NV),I=1,NV)$VN=NV
C.....CALCUL DES COMMUNAUTES.....
      DO101I=1,NV$H(I)=0.$DO101J=1,NF
  101 H(I)=H(I)+R(I,J)*R(I,J)
C.....NORMALISATION AVANT ROTATION.....
      DO103I=1,NV$DO103J=1,NF
  103 R(I,J)=R(I,J)/SQRT(H(I))
C.....DEBUT DES ROTATIONS VARIMAX.....
      CALLCALCV(NV,NF,V,R,S,KK)
      NG=NF-1
  335 VOLDCV=V
      DO385 M=1,NCST=M+1
      DO385N=J,NF
      E=BB=C=DTEMP=0.
      DO340I=1,NV
      S(I)=R(I,M)*R(I,M)-R(I,N)*R(I,N)
      A(I)=2.*R(I,M)*R(I,N)
      E=E+S(I)
      BB=BB+A(I)
      C=C+S(I)*S(I)-A(I)*A(I)
  340 DTEMP=DTEMP+S(I)*A(I)
      DD=2.*DTEMP
      XNUM=DD-((2.*E*BB)/VN)
      XDEN=C-((E*E-BB*BB)/VN)
C.....POSITIONNEMENT DE SIN/COS.....
      IF(XDEN)360,352,350
  350 FOURA=ATAN(XNUM/XDEN)$GOTO370
  352 IF(XNUM)354,356,358
  354 FOURA=-1.57$GOTO370
  356 FOURA=0.$GOTO370
  358 FOURA=1.57$GOTO370
  360 IF(XNUM)362,358,366
  362 FOURA=ATAN(XNUM/XDEN)-3.1416$GOTO370
  366 FOURA=ATAN(XNUM/XDEN)+3.1416
  370 ALP=FOURA/4.
      IF(ABS(ALP)-.01)385,385,375
  375 Z1=SIN(ALP)$Z2=COS(ALP)
      DO380I=1,NV
```

```
      B(M,M)=Z2$B(M,N)=-Z1$B(N,M)=Z1$B(N,N)=Z2
      SUM1=R(I,M)*B(M,M)+R(I,N)*B(N,M)
      SUM2=R(I,M)*B(M,N)+R(I,N)*B(N,N)
      R(I,M)=SUM1$R(I,N)=SUM2
  380 CONTINUE
  385 CONTINUE
      CALL CALCV(NV,NF,V,R,S,KK)
      IF(V-VCHECK-0.0001)390,390,385
C.....DENORMALISER LES LOADINGS APRES LES ROTATIONS.....
  390 DO400I=1,NV
      DO400J=1,NF
  400 R(I,J)=R(I,J)*SQRT(H(I))
      DO410J=1,NF$S(J)=A(J)=0.
      DO405I=1,NV
      IF(ABS(A(J))-ABS(R(I,J)))402,405,405
  402 A(J)=R(I,J)
  405 S(J)=S(J)+R(I,J)*R(I,J)
  410 S(J)=S(J)/VN
      PRINT293
  293 FORMAT(1H115X*ROTATIONS VARIMAX*/1X*PROPORTION DE VARIANCE EXPLIQUEE
     1EE PAR CHAQUE FACTEUR*)
      PPINT70,(I,S(I),I=1,NF)
   70 FORMAT(10(I4,F5.2,3X)/)
      PRINT294
  294 FORMAT(/1X*SATURATIONS MAXIMALES DES FACTEURS*)
      PRINT70,(I,A(I),I=1,NF)
      PRINT412
  412 FORMAT(/1X*COMMUNAUTES DES VARIABLES*)
      PRINT70,(I,H(I),I=1,NV)
      PRINT413
  413 FORMAT(1H11X*MATRICE APRES ROTATION*)
      CALL MATOUT(R,NV,NF,ND)
      RETURN$END
```

C A L C V

1. Mode d'appel

CALL CALCV (K, L, V, R, F, KL)

K : nombre de variables
L : nombre de facteurs
V : valeur de retour
R : matrice de structure factorielle à "rot..er"
F : vecteur de travail
KL: dimension de R et F dans le MAIN

2. Travail effectué

Calcule le critère de structure simple de Kaiser (rotations varimax). La structure simple sera atteinte quand V sera minimum.

3. Temps d'exécution

Négligeable

4. Mémoire requise

$93_{10} = 135_{B}$ mots

5. Sous-routines nécessaires

Aucune

6. Auteur

Inconnu, Univereité du Texas.

```
SUBROUTINE CALCV  TRACE                    CDC 6400 FTN V5.0-P556 OPT=0  73/1
      SUBROUTINE CALCV(K,L,V,R,F,KL)
      DIMENSIONR(KL,KL),F(KL)
      X=0.$DO5I=1,K$DO5J=1,L
    5 X=X+R(I,J)**4
      XK=K$X=XK*X$YI=0.
      DO15J=1,L$F(J)=0.$DO10I=1,K
   10 F(J)=F(J)+R(I,J)*R(I,J)
      F(J)=F(J)*F(J)
   15 YI=YI+F(J)
      V=X-YI$RETURN$END
```

B E T A

1. Mode d'appel

CALL BETA (NV, NF, NN, R, B, H, A)

NV: nombre de variables
NF: nombre de facteurs
NN: dimension des matrices et vecteurs dans le MAIN
R : matrice de corrélation (symétrique)
B : matrice des saturations factorielles
H : vecteur des communautés
A : vecteur de travail

2. Travail effectué

A partir de la matrice de corrélation R et de la matrice de structure factorielle B, la matrice des poids bêta est calculée et imprimée; la fidélité estimée des scores factoriels résultant de l'application de ces poids bêta est calculée par FIDFAC et imprimée.

3. Temps d'exécution

Pour 64 variables, environ 13 secondes par facteurs.

4. Mémoire requise

$171_{10} = 253_{8}$ mots

5. Sous-routines nécessaires

1) INVSYM

2) FIDFAC

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE BETA(NV,NF,NN,R,B,H,A)
      DIMENSION R(NN,NN),B(NN,15),H(NN),A(NN)
      NVA=NV+1
      DO3000L=1,NF
      DO2602I=1,NV
      R(I,I)=1.
      DO2602J=1,NV
 2602 R(J,I)=R(I,J)
      DO2603I=1,NV
 2603 R(NVA,I)=R(I,NVA)=B(I,L)
      R(NVA,NVA)=1.
      CALL INVSYM(DUM,NVA,R,A,NN)
      DO2604I=1,NV
 2604 B(I,L)=R(NVA,I)
 3000 CONTINUE
      DO3001I=1,NV$DO3001J=1,NF
 3001 R(I,J)=B(I,J)
      PRINT3002
 3002 FORMAT(1H110X*POIDS BETA*/)
      CALL MATOUT(R,NV,NF,NN)
      CALL FIDFAC(NV,NF,NN,B,H,A)
      RETURN$END
```

F I D F A C

1. Mode d'appel

CALL FIDFAC (NV, NF, ND, B, H, FID)

NV: nombre de variables
NF: nombre de facteurs
ND: dimensions de B, H et FID dans le MAIN
B : matrice des poids Bêta
H : vecteur des communautés
FID: vecteur de travail

2. Travail effectué

A partir de la matrice des poids Bêta (B) et du vecteur des communautés (H) pris comme estimé de la fidélité des items, FIDFAC calcule un estimé inférieur (lower bound) de la fidélité des scores factoriels et imprime les résultats.

3. Temps d'exécution

Négligeable

4. Mémoire requise

$126_{10} = 176_{8}$ mots

5. Sous-routines nécessaires

Aucune

6. Auteur

Richard Brunet, INRS-Education

```
UBROUTINE  FIDFAC    TRACE                          CDC 6400 FTN V5.0-P556 OPT=0  73/10

      SUBROUTINE FIDFAC(NV,NF,ND,B,M,FID)
      DIMENSION B(ND,ND),M(ND),FID(ND),SB(15)
C.....B = MATRICE DES POIDS BETA
C.....M= VECTEUR DES COMMUNAUTES
C..... FID= AURETOUR, LES FIDELITES
C..... ND= DIMENSION DES VECTEURS ET DE LA MATRICE
      DO3J=1,NF$FID(J)=SB(J)=0.$DO3I=1,NV
      SB(J)=SB(J)+B(I,J)*B(I,J)
    3 FID(J)=FID(J)+B(I,J)*B(I,J)*M(I)
      DO4J=1,NF
    4 FID(J)=1.-SB(J)+FID(J)
      PRINT5
    5 FORMAT(1H1,5X*FIDELITE DES SCORES FACTORIELS (ESTIMEE)*//)
      DO6I=1,NF
    6 PRINT7,I,FID(I)
    7 FORMAT(/10X*FACTEUR*I3*---*F5.2)
      RETURN$END
```

P R E R O T E

1. Mode d'appel

CALL PREROTE (NV, ND, R, S, A, NOPT)

NV: nombre de variables
ND: dimension de R, S et A dans le MAIN
R : matrice des composantes principales
S : vecteur des valeurs propres
A : vecteur de travail
NOPT: si 0, ne fait que positionner les fichiers
si 1, calcule ce qui est décrit plus loin

2. Travail effectué

1) lit sur tape 1 la matrice de corrélation, le vecteur de valeurs-propres et la matrice des composantes principales

2) écrit sur tape 2 la matrice des composantes principales

3) si NOPT $\neq$ 0, calcule le % cumulé de la trace et imprime pour chaque composante la valeur-propre et le % cumulé de la trace.

3. Témps d'exécution

Négligeable

4. Mémoire requise

$178_{10} = 262_8$ mots

5. Sous-routines nécessaires

1) MATOUT

2) PLOTTRA

6. Auteur

Richard Brunet, INRS-Education

```
      SUBROUTINE PREROTE(NV,ND,R,S,A,NOPT)
      DIMENSIONR(ND,ND),S(ND),A(ND)
      REWIND1
      READ(1)((R(I,J),J=1,NV),I=1,NV)
      READ(1)(S(I),I=1,NV)
      READ(1)((R(I,J),J=1,NV),I=1,NV)$VN=NV
      IF(NOPT.EQ.0)GOTO89
C.....VALEURS PROPRES CUMULEES.....
      A(1)=S(1)/VN
      DO2I=2,NV$J=I-1
    2 A(I)=S(I)/VN+A(J)
      PRINT3
    3 FORMAT(1H110X*VALEURS PROPRES*/10X*ITEM*10X*EIGENVALUE*10X*POURC. CU
     1CUM.*/)
      DO4I=1,NV
    4 PRINT5,I,S(I),A(I)
    5 FORMAT(I15,2F18.2)
      CALLPLOTIRA(S,NV,NDIM)
C.....IMPRESSION DES VECTEURS-PROPRES.....
      PRINT6
    6 FORMAT(1H110X*COMPOSANTES PRINCIPALES*)
      CALL MATOUT(R,NV,15,NDIM)
   89 WRITE(2)((R(I,J),J=1,NV),I=1,NV)
      RETURN$END
```

P L O T T R A

1. Mode d'appel

   CALL PLOTTRA (A, NV, ND)

   A : vecteur des valeurs-propres de la matrice de corrélation
   NV: nombre de variables
   ND: dimension de A dans le MAIN

2. Travail effectué

   Trace le "scree test" pour les composantes principales d'une matrice de corrélation (en termes de % de la trace).

3. Temps d'exécution

   Négligeable

4. Mémoire requise

   90_{10} = 132_8 mots

5. Sous-routines nécessaires

   Aucune

6. Auteur

   Richard Brunet, INRS-Education

```
SUBROUTINE  PLOTTRA   TRACE                          CDC 6400 FTN V3.0-P336 OPT=0  73/1
      SUBROUTINEPLOTTRA(A,NV,ND)$DIMENSIONA(ND),NFOR(3)
      DATANFOR(1),NFOR(3)/10H(/I8,1X1HI   ,7HX1H*I3)/
      PRINT2
    2 FORMAT(1H1//9X101(1H-))
      DO101I=1,NV$A(I)=A(I)/FLOAT(NV)
      NN=(100*(A(I)+0.005)-1)
      NX=NN+1$NN=NN*2
      IF(NN.GT.120)NN=120$IF(NN.LE.1)NN=1
      ENCODE(3,3,NFOR(2))NN
    3 FORMAT(I3)
   10 PRINTNFOR,I,NX
      PRINT4
    4 FORMAT(/9X101(1H-))
      RETURNSEND
```

C - Classification des sous-routines

1. Description du système de classification

Les sous-routines décrites dans la partie précédente et celles qui viendront seront classées dans le système qui suit.

A: sous-routines utilitaires

- A.1 Sous-routines de tri
- A.2 Sous-routines d'impression

B: statistiques élémentaires

- B.1 Moyenne, sigma, symétrie, voussure, etc
- B.2 Corrélations

C: nombres aléatoires

- C.1 Création de nombres aléatoires
- C.2 Création de nombres aléatoires avec contraintes

D: transformation des matrices

- D.1 Inversion
- D.2 Valeurs-propres et vecteurs-propres

E: analyse factorielle

- E.1 Rotations varimax
- E.2 Poids bêta
- E.3 Fidélité des scores factoriels
- E.4 Scree test

2. Classiffication des sous-programmes

A: sous-routines utilitaires

- A.1 Sous-routines de tri
  - A.1.1: tri d'un vecteur par ordre décroissant (ORDREVD)
  - A.1.2: tri d'un vecteur par ordre croissant (ORDREVC)
  - A.1.3: tri des lignes d'une matrice par ordre croissant (ORDMLC)
- A.2 Sous-routines d'impression
  - A.2.1: impression de matrices (MATOUT)

B: statistiques élémentaires

- B.1 Moyenne, sigma, sigma, $\sigma 1$, $\sigma 2$, médiane, etc
  - B.1.1: moyenne et sigma d'un vecteur (AMSIG)
  - B.1.2: moyenne, sigma, symétrie, voussure d'un vecteur (MOMENT)
  - B.1.3: médiane d'un vecteur (MEDIANE)
  - B.1.4: standardisation d'un vecteur (STANDA)

B.2 Corrélations
- B.2.1: calcul de la corrélation entre deux vecteurs (CORREL)
- B.2.2: transformation d'un R en Z de Fisher (FISHER)
- B.2.3: transformation d'un Z de Fisher en R (fonction inverse de B.2.2) (FISHBAK)

C: nombres aléatoires

C.1 Création de ces nombres
- C.1.1: initialisation au hasard de la suite de nombres aléatoires (INIRANF)
- C.1.2: création d'un vecteur de nombres aléatoires normaux standardisés (ALEAT)

C.2 Création de nombres aléatoires avec contraintes
- C.2.1: création d'un vecteur de corrélation donnée avec un premier vecteur, l'erreur étant distribuée normalement (SIMUL1)
- C.2.2: création d'un vecteur de corrélation donnée avec un premier, l'erreur étant distribuée rectangulairement (SIMUL2)

D: transformation des matrices

D.1 Inversion
- D.1.1: inversion d'une matrice symétrique (INVSYM)

D.2 Valeurs-propres et vecteurs-propres
- D.2.1: calcul des valeurs-propres et des vecteurs-propres d'une matrice (HOW)

E: analyse factorielle

E.1 Rotations varimax
- E.1.1: rotations varimax d'une matrice de composantes principales (VARIMAX)
- E.1.2: calcul du critère VARIMAX (CALCV)

E.2 Poids bêta
- E.2.1: calcul des poids bêta servant de la régression des variables sur les facteurs (BETA)

E.3 Fidélité des scores factoriels
- E.3.1: calcul de la fidélité des scores factoriels à partir des poids bêta et des communautés (FIDFAC)

E.4 Scree test
- E.4.1: impression du "scree test" (PREROTE, PLOTTRA)